A Learning-Based Assembly Framework for Contact-Intensive Tight-Tolerance Tasks

Bukun Son, Hyelim Choi, Donjun Lee†
Department of Mechanical Engineering, IAMD and IOER
Seoul National University
Seoul, South Korea
Email: sonbukun, helmchoi, djlee@snu.ac.kr

***Abstract*—We propose a two-stage framework that integrates a learning-based estimator and a controller to effectively manage contact-intensive tasks. The estimator utilizes a Bayesian particle filter combined with a Mixture Density Network (MDN) structure, which is proficient at solving non-injective issues from contact data. The controller merges self-supervised and reinforcement learning (RL) methods, separating the parameters of the low-level admittance controller into labeled and unlabeled parameters. To improve reliability and generalization capabilities, a transformer model is utilized in self-supervised learning. The suggested framework is validated on a bolting task using a precise real-time simulator and is successfully applied in an experimental setting.**

I. Introduction

Industrial automation relies on contact-rich tasks such as nut tightening, necessitating accurate object pose estimation and intricate control strategies. Most research, however, mainly focuses on one of these two components. The estimator minimizes larger uncertainties [5, 8], while the controller mitigates smaller real-time uncertainties [6]. A two-stage framework incorporating both components is essential for complex tasks.

For object pose estimation, vision-based methods are popular [5, 8], but suffer from relatively low accuracy due to factors such as low resolution, calibration errors, and most of all occlusions [6]. Contact sensing methods could overcome the issues, but produce multiple object poses from single contact info, a problem termed non-injectivity. This issue has been addressed by filtering sequential probabilistic information [3], but existing methods can be computationally demanding or require multiple unreliable and costly tactile sensors [11, 12]. Data-driven methods offer a potential solution for managing complex contacts with low computation costs [7].

Robotic assembly control strategy has significant challenges due to the need for adapting to complex and discontinuous contact effects [2, 4]. Reinforcement Learning (RL) has been commonly used to address these, but it is mainly applied to simple insertion tasks [4], limiting its effectiveness in more complex scenarios. [9] considers complex assembly tasks like screwing, but it is limited to validation in a simulated environment.

In response, we propose a two-stage framework for contact-intensive tight-tolerance assembly tasks in Fig. 1 handling complex contact geometry as shown. The framework consists of a learning-based estimator and controller, which is validated both in simulations and in real-world experiments. The estimator employs a particle filter with a mixture density network (MDN) structure for the Bayesian update to solve non-injectivity issues and calculate the estimation uncertainty. The controller leverages self-supervised learning and RL to improve reliability and data efficiency and use a transformer model to enhance generalization performance.

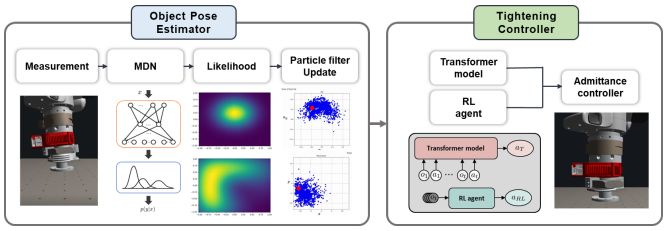

Fig. 1: The overall structure of the two-stage framework.

II. System Description

We construct the simulation and experimental setup with a robotic manipulator (Franka Emika Panda), an FT sensor (ATI gamma SI-65-6) to measure the 6-DOF contact wrench, and a HEBI X-series gripper capable of infinite rotation for rotational assembly tasks, as shown in Fig. 2. A manipulating object (e.g., nut) with the position $p_t \in \mathbb{R}^3$ and orientation $R_t \in SO(3)$ is rigidly attached to the HEBI gripper, and a fixed target object (e.g., bolt) with the position $p_t^{\text{tar}} \in \mathbb{R}^3$ and orientation $R_t^{\text{tar}} \in SO(3)$ is installed in the environment, where $\star_t$ represents a variable at time t. Motion planning and low-level control of the manipulating object are implemented in the 6-DOF Cartesian space. The low-level controller is an admittance controller with the reference manipulating object dynamics given as

$$M_t \ddot{e}_t + B_t \dot{e}_t + K_t e_t = F_t^c \tag{1}$$

where $e_t = [e_t^p, e_t^R]^T \in \mathbb{R}^6$ is the error vector, with the linear position error $e_t^p = p_t^{\text{ref}} - p_t \in \mathbb{R}^3$ and the orientation error as geometric error $e_t^R = \frac{1}{2}(R_t^T R_t^{\text{ref}} - R_t^{\text{ref}^T} R_t)^\vee$. Here, $p_t^{\text{ref}} \in \mathbb{R}^3$ is the reference position, $R_t^{\text{ref}} \in \text{SO}(3)$ is the reference orientation, and $M_t \in \mathbb{R}^{6\times6}$, $B_t \in \mathbb{R}^{6\times6}$, $K_t \in \mathbb{R}^{6\times6}$, $F_t^c \in \mathbb{R}^6$ are the inertia matrix, damping matrix, stiffness matrix, and external (contact) wrench, respectively. We set the inertia matrix M_t to be a constant M and the damping matrix B_t to be simplified as critically-damped $2\sqrt{MK_t}$.

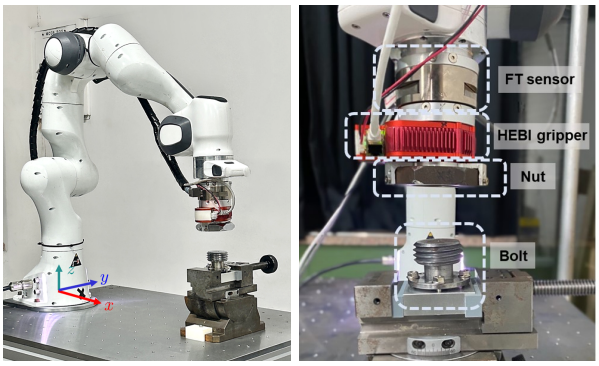

Fig. 2: The experiment environment setup consists of a Franka Emika Panda robotic manipulator, ATI Gamma FT sensor, HEBI X-series actuator, universal vice, nut, and bolt.

III. Data-driven Contact Pose Estimation

The contact pose estimator aims to determine the relative pose between two objects where contact may occur, addressing non-injectivity and predicting value uncertainty. This is achieved using an MDN-based probabilistic model and sequential Bayesian filtering. In our setup, the manipulating object's pose is exactly measured with manipulator kinematics, and the estimation is the pose error of the target object from the initial guess.

A. MDN pose probability model

MDN is trained to compute the conditional probability distribution of the target object pose error given the observation as $f(o_t) = P(p_t^{\text{tar}}, R_t^{\text{tar}}|o_t)$. f is the MDN and o_t is the observation. It assumes that the historical input provides more information than a single-step input, thereby the input of the MDN is defined as a history of observations in a sliding window as $o_t = [p_{t-n+1}, \log R_{t-n+1}, F^c_{t-n+1}, \dots, p_t, \log R_t, F^c_t]^T \in \mathbb{R}^{12n}$ to mitigate the partially observed problem, where $p_\star$ and $R_\star$ are the position and rotation of manipulating object and n is the window size ($n = 5$ in our application). Since we have no priors on the target object pose and observation, the output of the MDN, $p(p_t^{\text{tar}}, R_t^{\text{tar}}|o_t)$, can be considered as the likelihood as

$$\begin{aligned} P(p_t^{\text{tar}}, R_t^{\text{tar}}|o_t) &= \frac{P(o_t|p_t^{\text{tar}}, R_t^{\text{tar}})P(p_t^{\text{tar}}, R_t^{\text{tar}})}{P(o_t)} \\ &\propto P(o_t|p_t^{\text{tar}}, R_t^{\text{tar}}). \end{aligned} \tag{2}$$

In the case of a rotational fastening of a target object (e.g., bolt) installed on a table, it is necessary to estimate the horizontal position (xy plane) in Fig. 2 and a normal vector of the upper surface of the target object. Thus the output layer can be designed to represent four parameters consisting of x, y position error, and x, y coordinates of the axis normal vector error of the target object from the initial guess. The estimation network is constructed with four 1D convolution neural network (CNN) layers to ignore the time sequence dependency connected by an average pooling layer and two fully connected layers followed by the MDN layer consisting of 5 mixtures at the last.

B. Bayesian particle filter

A particle filter is utilized to predict the probability distribution of the target object pose error and estimate value uncertainty. This filter conducts Bayesian updates by incorporating sequential information and updating the posterior probability through the feed-forward likelihood calculation from the MDN model. In order to enhance robustness, particle weights are slightly adjusted after the update, and systematic resampling is employed when the effective sample size falls below a certain threshold. Gaussian white noise is introduced during resampling, and the filter continues to run until the weighted variance of the particles is below a specific threshold. The final estimate is obtained as the weighted sum of the particles.

C. Data collection

We collect training data for the estimation network in our real-time, physically accurate simulator [13] from a search motion where the manipulating object (e.g., nut) interacts with the target object (e.g., bolt) across 16,000 randomly sampled fixed target object poses. During the search motion, the manipulating object initially moves downward until the contact force surpasses a threshold ($F_z = 3$ [[]N] in our setup). Then, random searching motion is generated within the manipulator's velocity limit while maintaining contact. For each bolt pose, we simulate 2000 time steps (10 seconds at a frequency of 200 Hz), totaling 32 million steps. In addition, we randomize the friction coefficient to improve the sim-to-real transfer.

IV. Learning-based assembly controller

We optimize the parameters of the low-level controller to adapt to the residual error after the estimation step. In contact-intensive tasks requiring compliant behavior, we employ an admittance controller as the low-level controller. The controller parameters are divided into two categories based on whether they can be labeled or not. The reference pose can be labeled in the simulation since the exact bolt configurations can be measured. On the other hand, determining the optimal admittance gain or HEBI rotation velocity is not possible. To address this, we utilize supervised learning to learn the reference pose and employ RL to optimize the admittance gain and HEBI rotation velocity. Furthermore, by introducing the modulation wrench, F_t^{mod}, as an external force term, the error dynamics can converge to the desired objectives more rapidly. Consequently, the admittance controller dynamics (Eq. 1) is modified as

$$M_t\ddot{e}_t + B_t\dot{e}_t + K_t e_t = F_t^c + F_t^{\text{mod}} \tag{3}$$

A. Transformer-based self-supervised learning

The objective of self-supervised learning is to determine the reference pose, represented as ξ_{t+1}^{ref}, in the admittance dynamics. In the simulation, we can measure the exact pose of the bolt, enabling us to label the reference pose while

considering the bolt pitch. This facilitates efficient training using self-supervised learning. It is important to acknowledge that the problem of inputting contact wrenches presents a partially observed challenge. However, we cannot utilize the sequential filtering technique proposed in Section III since we need to predict a deterministic pose at each time step for the online controller. Fortunately, the relative errors during assembly are smaller compared to the estimation step, so we can mitigate this issue with a well-designed network structure. In our approach, we employ a transformer network structure, which has demonstrated high performance in sequential modeling by learning context through self-attention.

The token for the transformer is a nut pose $\xi_t = [p_t, q_t] \in \mathbb{R}^7$, contact wrench $F_t^c \in \mathbb{R}^6$, and reference pose $\xi_t^{\text{ref}} = [p_t^{\text{ref}}, q_t^{\text{ref}}] \in \mathbb{R}^7$ and the output action a_{Trans} is reference pose difference from the nominal trajectory for next step $[\Delta x_{t+1}, \Delta y_{t+1}, \Delta q_{w,t+1}, \Delta q_{x,t+1}, \Delta q_{y,t+1}, \Delta q_{z,t+1}] \in \mathbb{R}^6$. ($\Delta z$ is not necessary for the nut tightening task).

B. RL-based admittance controller

The objective of the RL-based controller is to optimize the unlabeled parameters, thereby improving the success rate during the tracking of the predicted desired pose. The reference pose predicted by the transformer network may contain errors compared to the true pose obtained from the actual bolt pose, which can lead to fastening failure or jamming. The optimization variables are K_t and F_t^{mod} Eq. 3, and w_{HEBI} for HEBI rotation. To address the partially observed issue, the observation $o_t = [p_{t-T}, q_{t-T}, Fc_{t-T}, \ldots, p_{nut,t}, q_{nut,t}, Fc_t]$ includes the history of the nut pose and contact wrench. This helps mitigate the challenges associated with partial observation. For training, we utilize the proximal policy optimization (PPO) algorithm [10], which is well-suited for continuous robot actions and offers high data efficiency.

V. Simulation and Experiments

The proposed two-stage framework has been developed and validated using the M48 bolt-nut scenario in KSB-0201 [1]. The initial bolt error range assumed $12\,[\text{mm}]$ in both the x and y directions of position, as well as $10\,[\text{deg}]$ in orientation allowing for rough contact between objects, which is larger than what has been used in prior studies.

The likelihood model based on MDN represents a probability distribution that includes the true values, in contrast to deterministic estimation, as depicted in Fig. 4. The filtering process is terminated when the variance of the particles becomes smaller than a specified threshold of 10^{-3}. The final estimation error is computed as the 2-norm error between the ground truth and the estimated pose, which is defined as the weighted average of the particles. In the simulation, through 20 trials with randomly initialized bolt poses, we observe that the mean final position error is $1.60\,[\text{mm}]$ (with a standard deviation of 1.30), and the mean orientation error is $2.60\,[\text{deg}]$ (with a standard deviation of 1.81). The final estimation error is significantly reduced compared to the initial error, with a reduction of 78.58% in the mean position error and 44.21% in the mean orientation error. The average time required for the estimation process is 4.49 seconds. In the experiment with the same process, we observe that the mean final position error is $3.80\,[\text{mm}]$ and the mean orientation error is $3.71\,[\text{deg}]$.

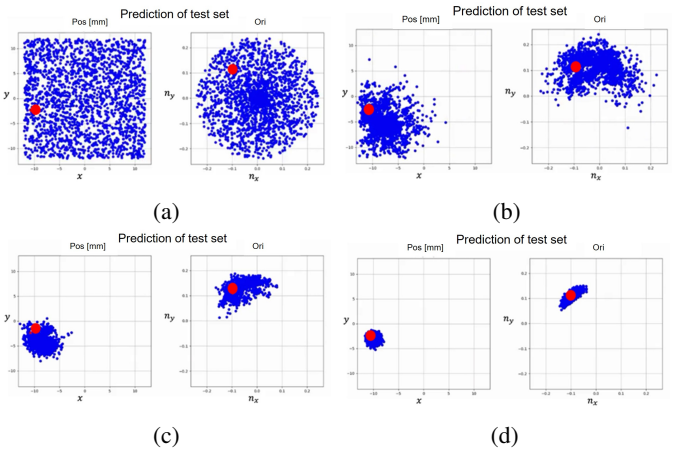

Fig. 3: The sequential particle filter results from (a) to (d) in time order. The blue dots are the particles and the red dots mark the true object pose offsets for the position (left) in mm and the orientation (right) in each stage.

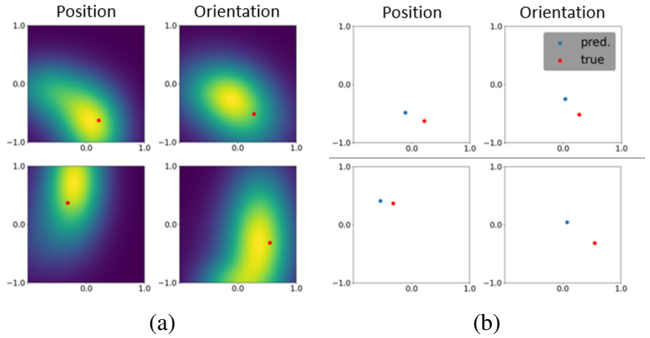

Fig. 4: The comparison of the probabilistic estimation network and the deterministic estimation network (estimation values are normalized). (a) Estimated probability density functions of the target object pose (b) The plots of a deterministic estimation from a simple neural network.

The assembly controller conducted arbitrary sampling of bolt poses with a position error of $5\,[\text{mm}]$ and an orientation error of $5\,[\text{deg}]$. This allowed for a margin from the maximum error during the estimation step. Quantitative metrics were evaluated based on the position and orientation errors relative to the target completion of two turns after fastening, as well as the success rate based on the fastenings of one turn or more. To evaluate the validity of the transformer model, the model structure baselines considered were the fully connected model and the LSTM model. For the baseline of the entire assembly algorithm, we considered a nominal trajectory created under the assumption that there are no pose errors in the bolt and a random trajectory generated by random sampling within the range of the action space at each step. In the simula-

	Pos err [mm]	Ori err [deg]	Succ rate [%]
FC	2.32	0.73	68
LSTM	2.67	0.71	62
Nominal	4.57	1.56	63
Random	3.96	1.23	76
Transformer	1.40	0.37	81
Transformer + RL	**0.61**	**0.28**	**98**

TABLE I: The result of the assembly in the simulation.

	Pos err [mm]	Ori err [deg]	Succ rate [%]
Nominal	5.61	1.46	52
Transformer	3.99	0.51	88
Transformer + RL	**2.59**	**0.58**	**98**

TABLE II: The result of the assembly in the experiment.

tion, as shown in Table I, the transformer exhibited higher performance compared to the fully connected and LSTM models. Furthermore, even when using the transformer alone, it demonstrated higher performance compared to the nominal trajectory and random trajectory. Combining the transformer with the RL model resulted in a remarkably high success rate of 98%. The experiments were conducted in three scenarios for safety considerations: nominal trajectory, transformer only, and transformer combined with RL. As shown in Table II, similar to the simulation, the transformer with RL showed a 98% success rate with significantly lower final position and orientation errors. The proposed algorithm has demonstrated a high success rate, achieving close to 100% not only in simulations but also in experiments. This marks a significant achievement in industrial automation.

VI. Conclusion

In conclusion, the suggested two-stage framework for contact-intensive tight-tolerance assembly tasks is successfully implemented. The framework, which integrates a learning-based estimator and a controller, is validated through simulations and real-world experiments. The results from the simulations and experiments demonstrated a significant reduction in both position and orientation errors with almost 100% success rate, highlighting the effectiveness of the proposed framework. Future work could focus on enhancing the framework's performance and exploring its applicability to a broader range of tasks and objects.

VII. Acknowledgments

This research was supported by the Industrial Strategic Technology Development Program (20001045, 20008957) and the Institute of Information & Communications Technology Planning & Evaluation (2021-0-00896).

References

[1] "Metric coarse screw threads," KS B 0201, Korean Standard, 2016.

[2] Cristian C Beltran-Hernandez, Damien Petit, Ixchel G Ramirez-Alpizar, and Kensuke Harada. Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach. *Applied Sciences*, 10 (19):6923, 2020.

[3] Siddharth R Chhatpar and Michael S Branicky. Particle filtering for localization in robotic assemblies with position uncertainty. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3610–3617. IEEE, 2005.

[4] Todor Bozhinov Davchev, Kevin Sebastian Luck, Michael Burke, Franziska Meier, Stefan Schaal, and Subramanian Ramamoorthy. Residual learning from demonstration: Adapting dmps for contact-rich manipulation. *IEEE Robotics and Automation Letters*, 2022.

[5] Xinke Deng, Junyi Geng, Timothy Bretl, Yu Xiang, and Dieter Fox. icaps: Iterative category-level object pose and shape estimation. *IEEE Robotics and Automation Letters*, 7(2):1784–1791, 2022.

[6] Guoguang Du, Kai Wang, Shiguo Lian, and Kaiyong Zhao. Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review. *Artificial Intelligence Review*, 54(3):1677–1734, 2021.

[7] Shiyu Jin, Xinghao Zhu, Changhao Wang, and Masayoshi Tomizuka. Contact pose identification for peg-in-hole assembly under uncertainties. In *2021 American Control Conference (ACC)*, pages 48–53. IEEE, 2021.

[8] Lakshadeep Naik, Thorbjørn Mosekjær Iversen, Aljaz Kramberger, Jakob Wilm, and Norbert Krüger. Multi-view object pose distribution tracking for pre-grasp planning on mobile robots. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[9] Yashraj Narang, Kier Storey, Iretiayo Akinola, Miles Macklin, Philipp Reist, Lukasz Wawrzyniak, Yunrong Guo, Adam Moravanszky, Gavriel State, Michelle Lu, et al. Factory: Fast contact for robotic assembly. *arXiv preprint arXiv:2205.03532*, 2022.

[10] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[11] Andrea Sipos and Nima Fazeli. Simultaneous contact location and object pose estimation using proprioception and tactile feedback. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3233–3240. IEEE, 2022.

[12] Felix von Drigalski, Kennosuke Hayashi, Yifei Huang, Ryo Yonetani, Masashi Hamaya, Kazutoshi Tanaka, and Yoshihisa Ijiri. Precise multi-modal in-hand pose estimation using low-precision sensors for robotic assembly. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 968–974. IEEE, 2021.

[13] Jaemin Yoon, Minji Lee, Dongwon Son, and Dongjun Lee. Fast and accurate data-driven simulation framework for contact-intensive tight-tolerance robotic assembly tasks. *arXiv preprint arXiv:2202.13098*, 2022.